Charles Keer

ME311

Module 5 Assignment

4/4/2022

Task 1A:

Test Image	Actual Image	Model Classification	Classification Confidence (%)	Correct Classification?
TestImage1	Spiderman	Captain America	80	NO
TestImage2	Spiderman	Spiderman	94	YES
TestImage3	Spiderman	Captain America	85	NO
TestImage4	Spiderman	Spiderman	100	YES
TestImage5	Spiderman	Spiderman	97	YES
TestImage6	Hulk	Captain America	64	NO
TestImage7	Hulk	Hulk	100	YES
TestImage8	Hulk	Hulk	100	YES
TestImage9	Hulk	Hulk	99	YES
TestImage10	Hulk	Hulk	99	YES
TestImage11	Captain America	Captain America	99	YES
TestImage12	Captain America	Captain America	100	YES
TestImage13	Captain America	Captain America	97	YES
TestImage14	Captain America	Captain America	99	YES
TestImage15	Captain America	Captain America	100	YES

Shareable link: https://teachablemachine.withgoogle.com/models/c8u-aIPmk/

Task 1B:

There are a few Spiderman images that were not identified correctly. This can be for several reasons. One reason is that there are not a lot of reference images for Spiderman compared to Captain America, who shares a similar look. This means the model has less information to refer to Spiderman when testing. Another problem that the model had was bad lighting and strange/partial pictures of the subjects. With the model under bad lighting, the colors seem to blend together and the model has difficulty determining the correct class.

Task 2:

Shareable Link: https://teachablemachine.withgoogle.com/models/OeDOOXMTn/

The way that this model seems to work is through shape and color recognition. Therefore, it is important to include training images that are not greyscaled and do not include details of multiple classes. For example, a Spiderman image that is black and white would not be a good choice because red and blue would not be visible. Another example would be the pictures where multiple classes were in the image, such as Hulk and Spiderman next to each other. This would cause the model to recognize some of the Hulk characteristics as Spiderman and vice versa. This model was tested by inputting database 1 into the model as well as downloading several internet images of all three characters. The model was able to predict it to a reasonable accuracy, leading me to believe that this method of choosing images is correct.

Task 3:

To be able to control the robot through a webcam, there would need to be a lot of training needed. The first step would be to determine which hand movements would correspond to different robot actions. For example, a closed fist can be mapped to make the robot stop moving. Once these gestures are determined, the neural network can start to be trained. For a reliable neural network, there would need to be several thousand images for each gesture in different orientations to get as complete of a mapping as possible. Once the neural network is trained, the outputs of the model would correspond to different Arduino/Romi functions. After this, the robot can be tested and necessary changes made.

Pseudocode:

```
-Include all necessary libraries needed to connect to the model
-Define all variables
-Setup Loop:
    -Initialize all necessary robot functions (like motors)
-Void Loop:
    -Take input from model
    -Use a Switch syntax:
    -Case: “STOP”
        -Set motors to zero
    -Case: “Drive Forward”
        -Set both motors to forward
    -Case “Drive Backward”
        -Set both motors to reverse
    -Case “90Right”
```

```
        -Turn the left wheel counterclockwise and the right wheel clockwise. Utilize the onboard IMU for accurate degrees.
    -Case “90Right”
        -Turn the left wheel clockwise and the right wheel counterclockwise. Utilize the onboard IMU for accurate degrees.
    -Delay for 1000 ms to give the model enough time to respond between loops

End
```